4-30-2007

A Multi-Camera System for a Real-Time Pose Estimation

Andreas Savakis
Rochester Institute of Technology

Matthew Erhard
Rochester Institute of Technology

James Schimmel
Rochester Institute of Technology

Justin Hnatow
Rochester Institute of Technology

Recommended Citation
Andreas Savakis, Matthew Erhard, James Schimmel, Justin Hnatow, "A multi-camera system for real-time pose estimation", Proc. SPIE 6560, Intelligent Computing: Theory and Applications V, 656006 (30 April 2007); doi: 10.1117/12.719633; https://doi.org/10.1117/12.719633

A Multi-Camera System for Real-Time Pose Estimation

Andreas Savakis, Matthew Erhard, James Schimmel and Justin Hnatow

Department of Computer Engineering, Rochester Institute of Technology, Rochester, NY 14623

ABSTRACT

This paper presents a multi-camera system that performs face detection and pose estimation in real-time and may be used for intelligent computing within a visual sensor network for surveillance or human-computer interaction. The system consists of a Scene View Camera (SVC), which operates at a fixed zoom level, and an Object View Camera (OVC), which continuously adjusts its zoom level to match objects of interest. The SVC is set to survey the whole filed of view. Once a region has been identified by the SVC as a potential object of interest, e.g. a face, the OVC zooms in to locate specific features. In this system, face candidate regions are selected based on skin color and face detection is accomplished using a Support Vector Machine classifier. The locations of the eyes and mouth are detected inside the face region using neural network feature detectors. Pose estimation is performed based on a geometrical model, where the head is modeled as a spherical object that rotates upon the vertical axis. The triangle formed by the mouth and eyes defines a vertical plane that intersects the head sphere. By projecting the eyes-mouth triangle onto a two dimensional viewing plane, equations were obtained that describe the change in its angles as the yaw pose angle increases. These equations are then combined and used for efficient pose estimation. The system achieves real-time performance for live video input. Testing results assessing system performance are presented for both still images and video.

1. INTRODUCTION

Facial pose estimation is important in image understanding and activity recognition, as faces often appear in non-frontal position. Pose estimation introduces a way to detect the head direction, so that this information can be used to enhance further processing or provide information for activity recognition. A face can undergo pitch, roll and/or yaw rotations, which makes operations such as face recognition more difficult. The 'yaw' of a face is the degree by which the face turns horizontally to the left or right. The yaw angle has the greatest possible benefit to applications, since it is an indication of which way a person is looking or heading. This paper deals with yaw angle estimation in real time within a network of camera sensors. A geometric head model approach is presented where proportions between the facial features are used to calculate the pose angle. This provides a good tradeoff between accuracy and computational efficiency.

An important aspect of pose estimation methods is determining which human facial features to use. The use of the eyes as feature points is popular, because eyes are prominent and relatively easy to detect. On the other hand, the nose is harder to detect due to its lack of contrast, however, some methods [1, 2] make use of this feature with proper filtering. It should be kept in mind that facial feature detection is often the most computationally demanding portion of a pose estimation algorithm [3]. In our approach, we limit the number of features to three (left eye, right eye and mouth), which reduces complexity without compromising accuracy. The proposed method employs Artificial Neural Networks (ANNs) for feature detection and exploits the angles of a geometric template formed by the detected features. Other methods that have used ANNs are mainly appearance-based [4] [5]. These methods train their networks

Intelligent Computing: Theory and Applications V, edited by Kevin L. Priddy, Emre Ertin,
Proc. of SPIE Vol. 6560, 656006, (2007) · 0277-786X/07/$18 · doi: 10.1117/12.719633

on small images of the whole face turned at different angles in contrast to the proposed method which searches for individual facial features.

The template used for this paper is similar to other triangle or pyramidal methods presented in [2] and [6]. These methods perform pose estimation using either a classification system based on template deformation or orthogonal projections of the detected points. Our method focuses on efficient pose estimation in real time by using a small number of feature points and by directly determining pose from the angles of the template. Facial expression is assumed to be neutral with the eyes open. Finally, there are no occlusions, such as hats, glasses, facial hair, and scarves that may prevent the detection of facial features. The next section overviews the multi-camera environment used for pose estimation. Section 3 outlines the pose estimation approach and the methods used for facial feature detection. Results and conclusions are presented in Sections 4 and 5 respectively.

2. REAL TIME MULTI-CAMERA ENVIRONMENT

2.1 System Setup

Pose estimation may be performed on still images or on video data, e.g. video streams in a multi-camera visual sensor network environment that operates in real-time. There are various configurations in which multiple cameras can be used in order to achieve automated video understanding. Most systems utilize distributed cameras with non-overlapping fields of view. Carnegie Mellon's Video Surveillance and Monitoring project used a distributed network of active video sensors [7]. In CMU's surveillance system, many cameras are used together to identify and track objects as they move between the fields of view of the cameras. Each of the cameras is connected to a processor and has the ability to detect and track objects on its own. In a hierarchical sensor slaving configuration, one camera and associated processing unit acts as the master and the other sensors act as slaves. The master unit has a wide-angle view of the area and is able to track all objects within that area. The real world coordinates of the object(s) tracked by the master are passed to the slave(s), which zoom in to acquire more detailed images of the object(s). In this paper, a Scene View Camera (SVC), acts as the master and detects faces in the scene. An Object View Camera (OVC) zooms into the face of interest and performs pose estimation after detecting the relevant facial features.

2.2 System Components

The system flow includes modules that perform the following operations: skin detection, face detection, camera correspondence, facial feature detection and pose estimation. The location of facial candidate regions is based on color segmentation obtained by performing skin detection. The Hue Saturation Value (HSV) color space is used for skin detection because it provides good concentration of skin tones and processing in this space is simple and effective [6,8]. Efficiency was increased by creating a trainable lookup table which is used to determine whether a pixel's color is a skin tone. Each frame is processed using color segmentation and the resulting image is examined for regions with sufficient numbers of skin pixels to become a face candidate.

Face candidate regions are evaluated using a Support Vector Machine (SVM) classifier. SVMs have been shown to be effective for pattern recognition dealing with two-class problems [9] and can be trained to distinguish faces from non-faces. Each face candidate is converted to a grayscale image, undergoes histogram equalization, and is scaled to a standard size. The normalized image is classified by the SVM and the face with the highest match is selected as the prominent face in the scene.

Once the most prominent face has been identified, the OVC is used to obtain facial data at a higher resolution. The location of the most prominent face is calculated and the OVC is given the proper pan

and tilt angles, as well as the new zoom level. The face detection algorithm is then run on the images the OVC captures and the facial region of interest (ROI) is passed to the feature extraction method. Regions of interest for the two eyes and the mouth are found. Using the neural networks on the entire face image would be too computationally intensive for the system to be able to run in real time. In order to speed up the processing time, the general location of the eyes and mouth are found based on face shading. Given a contour map of the face, the eyes and mouth generally correspond to the lowest/darkest locations. Therefore, by searching for minima and maxima, it is possible to locate the feature regions without extensive processing. The neural networks are then used exclusively on these feature regions to determine the locations of the eyes and the mouth. The output responses of the ANNs are used to generate activation maps for each of the facial features. The activation maps are smoothed via a Gaussian filter and the locations with the highest responses are selected as the facial feature locations. Knowledge of the eye locations helps identify roll in the subject's posture. The yaw of the face can be determined by measuring the angles of the triangle formed by connecting the center of the eyes and mouth of the subject's face image. The pose estimation method is presented next.

3. POSE ESTIMATION

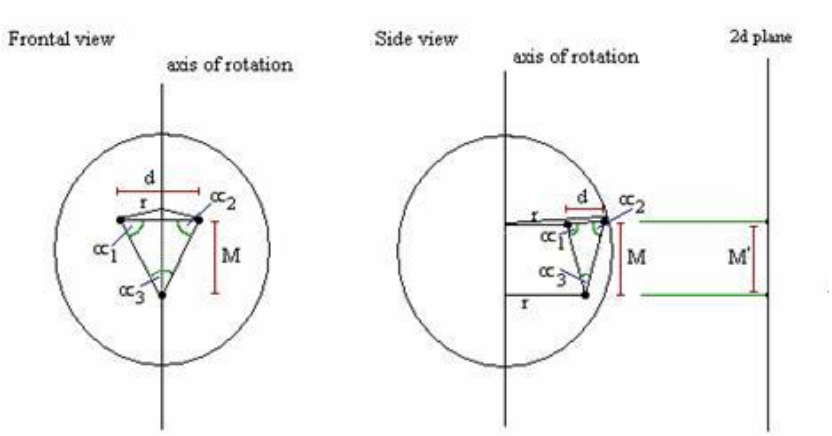

Figure 1. Geometric Head Model used for Pose Estimation.

A geometric approach is pursued for pose estimation, as described in [10]. The head model used for pose estimation is shown in Figure 1, where the eyes and mouth are used as the primary feature points. The head itself is treated as a spherical object of radius *r* which rotates upon the y-axis. The mouth and eyes are treated as a vertical plane on this sphere. The distance between the eyes is labeled *d*. By projecting the model of the head onto a two dimensional plane, we can develop equations which show the change in the angles as the model increases in yaw pose angle.

$$\alpha_1'' = \arctan\left(\tan(\alpha_1) * \frac{\sin\theta_0}{\sin(\theta_T) - \sin(\theta_T - \theta_0)}\right)$$

$$\alpha_2'' = \arctan\left(\tan(\alpha_2) * \frac{\sin\theta_0}{\sin(\theta_0 - \theta_T) - \sin(\theta_T)}\right)$$

$$\alpha_3'' = \pi - (\alpha_1'' + \alpha_2'')$$

In the above equations, the pose angle is represented by θ_T, variables α_1, α_2 and α_3 represent the angles of the face in the frontal position, and θ_0 represents the angle formed by the eyes and the axis of rotation. The quantities $\theta_0, \alpha_1, \alpha_2$ and α_3 all vary from individual to individual, and make it necessary to normalize the angle relative to one another when performing the final pose estimation.

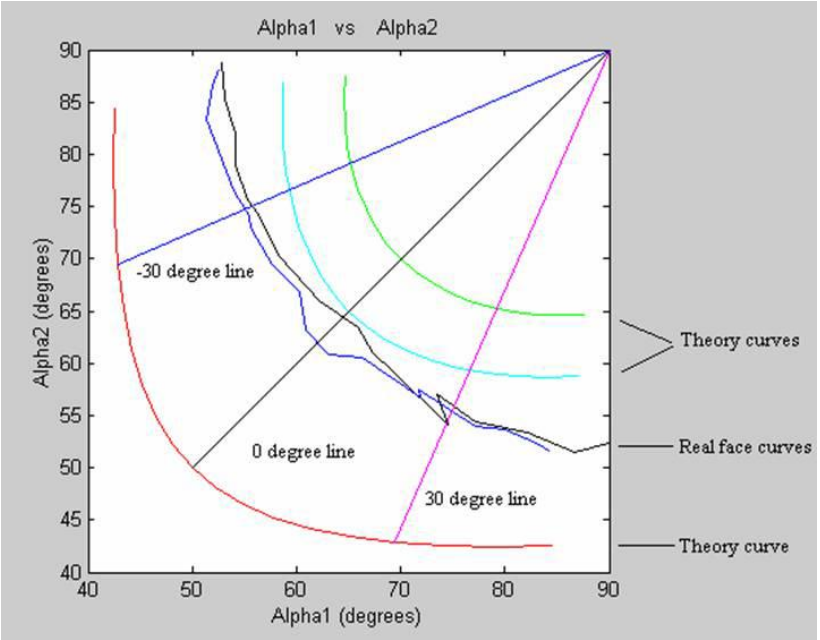

Figure 2. Plot of angles α_1 versus α_2 using theoretical and real data.

Since the sum of the angles in a triangle is always equal to π, only two of the angles in the pose triangle are independent. Figure 2 is a graph of α_1 versus α_2, the two top angles of the triangle, as the face changes pose. The curves represent three theoretical curves and two curves generated from image sets. The lines from the origin connect points on the curves at -30°, 0° and 30°. This graph helps illustrate how the pose angles form curves around a given focal point. This is due to the symmetry inherent in the human face. At the forward position these two angles are assumed to be equal. As the head changes direction the difference between these angles increases.

The pose angle is equal to the angle formed by the center line, where the face is in the frontal position, and the line formed by the calculated α_1, α_2 point and the origin. Using this idea and the previous equations, a simplified equation for the pose estimate is obtained:

$$\theta_T = \arctan\left(\frac{\alpha_2 - \alpha_1}{\alpha_3}\right)$$

The above equation provides a computationally efficient way of estimating pose angle. Detection of the facial features that define the pose triangle is the most computationally demanding task. The next section presents pose estimation results for still images and real-time video data processed within a multi-camera environment.

4. RESULTS

4.1 Performance of Feature Detectors

Before presenting pose estimation results, the performance of the feature detectors is summarized. The performance of the feature detectors is critical, because pose estimation heavily depends on these facial features. The SVM face detector was based on a polynomial kernel and was trained using 385 images. Training images included 150 face images from 20 subjects and 235 images of non-faces, consisting of hands, arms and background. The trained SVM classifier consisted on 245 support vectors and was tested on 200 images. It achieved 96% accuracy for face images and 98% accuracy for non-face images.

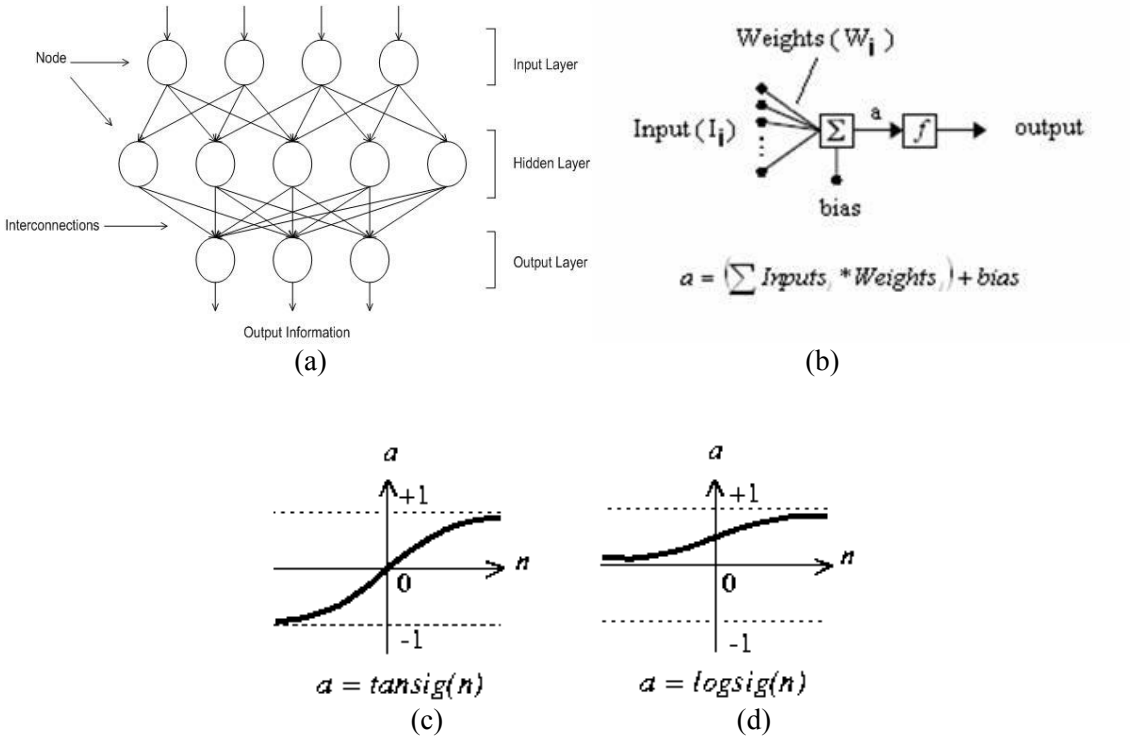

Figure 3. (a) Neural Network Topology; (b) Single Neuron Configuration;
(c) Tan-Sigmoid Transfer Function; (d) Log-Sigmoid Transfer Function.

An eye detector neural network, shown in Figure 3, was developed to detect both left and right eyes of a subject. It was trained using 2760 eye images and 13800 non-eye images. The ANN contains 231 input nodes, 14 hidden layer nodes using a Tan-Sigmoid transfer function, and one output node using the Log-Sigmoid transfer function. The eye network performed at 90.84% accuracy for eye images and 91.53% for non-eye images.

The mouth network neural network was trained to detect neutral expressions, but can effectively detect mouths with some variation in expressions. It was trained using 1430 mouth images and 7150 non-mouth images. The network contains 429 input nodes, 13 hidden layer nodes using a Tan-Sigmoid transfer function, and one output node using the Log-Sigmoid transfer function. The mouth network performed at 93.26% accuracy for eye images and 93.75% for non-eye images.

4.2 Pose Estimation Results on Still Images

In order to test the pose estimation algorithm on still images, an image database was generated under controlled conditions. Images of ten individuals were obtained at various poses taken at 5° increments from 30° to -30°, with 0° being considered the frontal position. None of these images were used in the training of the ANNs. Lighting, facial expression, and head movement were kept constant.

Initially, to determine the upper limit of the method's performance, the center locations for the right eye, left eye and mouth were selected manually. From these values the pose was calculated on each of the test images. With most image sets the error tends to be lowest at the frontal position and worsens gradually as the head approaches the extreme angles. Table 1 summarizes the average error from all ten test sets was 4.28 degrees.

Average Pose Estimation Error (degrees)	
Manual detection	Automatic detection
4.28	6.41

Table 1: Average Pose Estimation Error over 10 Subjects.

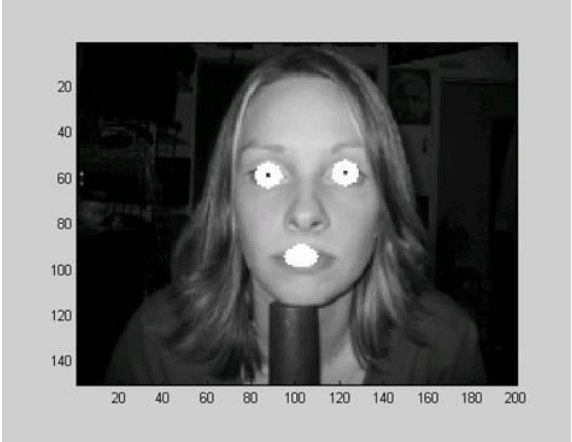

Figure 4. Example of Facial Feature Detection of a Still Image.

Results based on automatic feature detection were obtained using all 130 images. The algorithm was able to detect the features and apply a template correctly for 120 of the images, which corresponds to a 92.3% correct detection rate. An example of feature detection on a still image is shown in Figure 4. Problems with the automatic scaling or a person's particular features caused most of the missed detection. The automatic detection method performed well compared to the best possible performance obtained by manual methods, as shown in Table 1. From this table the increase in error due to automatic feature detection was about 2.13°, which is acceptable.

4.3 Real-Time Pose Estimation Results in a Multi-Camera System

The real time system consisted of four cameras configured in pairs, where two of the cameras acted as Scene View Cameras with non-overlapping fields of view, and the other two were Object View Cameras. Each SVC and OVC camera pair operated in a master-slave configuration and was connected to a workstation where processing took place. The two workstations communicated over Ethernet connection to alert each other about the presence and location of faces.

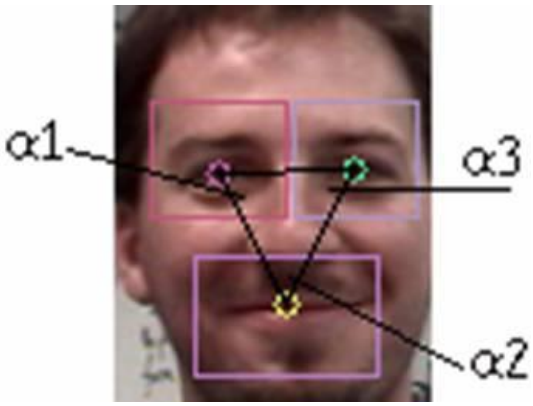

Figure 5. Example of Real-Time Facial Feature Detection on a Video Frame.

After the face detection system is run on the image captured by the SVC, the location of the face along with the size of the encompassing ROI is passed onto the camera view correspondence system. The face width, which is estimated over five frames, was used to estimate the distance between the subject and the camera. The offset of the subject's ROI in the SVC window is used to estimate the angle the SVC would need to pan in order to center the subject. This angle and the estimated distance determine the subject's location. The subject's location along with the SVC's and OVC's physical location is used to compute how the OVC must turn to see the same face the SVC sees. This requires finding the new pan, tilt, and zoom needed by the OVC. The images obtained by the OVC are processed through the face detection subsystem and the ROI containing the face is passed to the pose estimator. Figure 5 illustrates an example of facial feature detection on a video frame that is performed in real time.

Subject	Minimum Error (deg)	Maximum Error (deg)	Average Error (deg)
1	0.42	8.20	3.30
2	0.87	11.95	5.41
3	0.63	24.03	6.47
4	0.01	11.44	3.73
5	0.37	6.27	3.56
6	1.48	17.14	7.81
7	0.01	10.38	5.11
8	0.02	8.13	3.50
9	0.56	12.35	5.04
10	1.16	16.13	6.39
Average	0.55	12.60	5.08

Table 2. Average Real-Time Pose Estimation Error of All Subjects

To test the pose estimator in real time, ten subjects were placed 110 cm away and 3 cm offset from the SVC and were asked to look at targets located at 5 degree intervals between +25° and -25°. Table 2 shows the minimum, maximum, and average error in the estimated pose for ten subjects. Since only two of the tested subjects had valid pose estimates at the -25 and 25 degree marks, the data in Table 2 is based only on the estimates gathered from -20° to 20°. The minimum error results show that each subject had at

least one reading that was very accurate. This generally occurred either at the -5°, 0, or 5° reading. Overall the average pose error was 5.08°, which is comparable to the average error obtained for pose estimation in still images. As the pose angle increased the estimate generally got worse due to errors in the feature detection. The source of this error due to roll in the subjects' head or improper location of the features because of changes in angle and/or shading.

Ideal Pose	Valid Readings (Max=10)	Avg. Pose	Avg. Error
-25	1	-18.87	6.13
-20	4	-9.01	13.93
-15	8	-8.47	8.07
-10	10	-6.66	5.37
-5	10	-6.13	1.85
0	10	-0.46	4.37
5	10	5.81	2.86
10	10	9.19	3.05
15	9	10.78	5.01
20	6	11.59	8.41
25	1	20.08	4.92

Table 3. Average Real-Time Pose Estimate and Pose Estimation Error

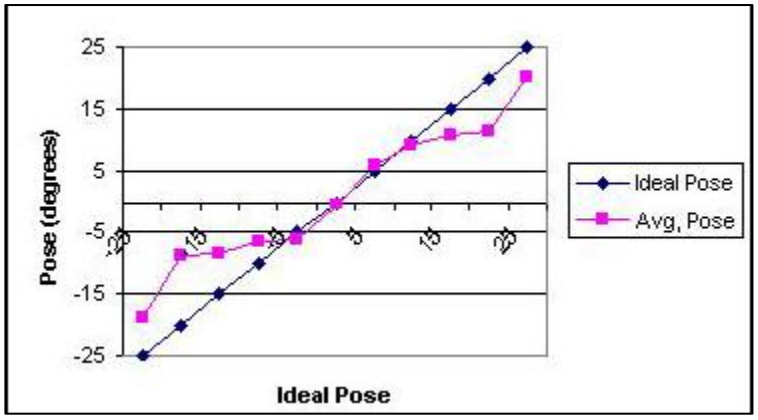

Figure 6. Actual (Ideal) Pose versus Estimated Pose

Table 3 shows the numbers of subjects with valid readings at each pose angle, as well as the average pose estimation and pose error at each angle. A valid reading is one in which the face and the facial features were detected, which allowed for the pose to be estimated. As shown, the system was able to capture the pose of nearly all of the subjects within the -15 to 15 degree range. The reason that a few subjects didn't have their pose captured was because the feature detector failed to detect one or more of their features. Beyond ±15 degrees fewer and fewer readings were captured because either the features or the face could not be found by one of the cameras. In addition, the quality of the estimation generally drops as the magnitude of the pose angle increases. At the ±25 degree marks, only one of the subjects had

both their face and features detected, so the error at these points is not as accurate as the others. Figure 6 illustrates that in most cases the pose estimation falls shallow of the actual value.

5. CONCLUSIONS

This paper presents a robust and efficient pose estimation method that is suitable for real time implementation. The accuracy of the algorithm makes this method capable of automatically determining the pose of a human face without requiring an excessive number of computations that must be performed on the image. Currently this method is designed to determine pose in only the yaw direction. Future work should expand the model to include ways of determining changes in the roll and the pitch of the head.

ACKNOWLEDGMENTS

This research was funded in part by the Eastman Kodak Company and the Center for Electronic Imaging Systems (CEIS), a NYSTAR-designated Center for Advanced Technology in New York State.

REFERENCES

[1] M. C. Burl and P. Perona, "Recognition of Planar Object Classes," *IEEE Conf. Computer Vision and Patten Recognition*, San Francisco, CA, June 1996.

[2] K. N. Choi, M. Carcassoni, and E. Hancock, "Estimating 3D Facial Pose using the EM Algorithm." *9th British Machine Vision Conference,* Southampton, UK, August 1998.

[3] R. Fergus, P. Perona, and A. Zisserman, "Object Class Recognition by Unsupervised Scale-Invariant Learning," *IEEE Conference on Computer Vision and Pattern Recognition*, Madison, WI, 2003

[4] H. A. Rowley, S. Baluja, and T. Kanade, "Neural Network Based Face Detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. 20, No. 1, pp. 23-38, 1998.

[5] J. Haddadnia, K. Faez, and M. Ahmadi. "N-Feature Neural Network Human Face Recognition", *15th International Conference on Vision Interface*, Calgary, Canada, May 2002

[6] A. Yilmaz and M. Shah, "Automatic Feature Detection and Pose Recovery for Faces," *The 5th Asian Conference on Computer Vision*, Melbourne, Australia, January 2002.

[7] H. Fujiyoshi, R. Collins, T. Kanade, and A. Lipton, "Algorithms for cooperative multisensor surveillance," *Proceedings of the IEEE*, Vol. 89, No. 10, pp. 1456 - 1477, October, 2001.

[8] M. Storring, H. J. Andersen and E. Granum "Skin Colour Detection under Changing Lighting Conditions," *7th symposium on Intelligent Robotics Systems*, Coimbra, Portugal, July 1999.

[9] N. Cristianini, and J. Shawe-Taylor, *An introduction to Support Vector Machines (and other kernel-based learning methods)*, Cambridge University Press, Cambridge, UK, 2000.

[10] A. Savakis and J. Schimmel, "Facial Pose Estimation for Image Retrieval," *VISAPP 2007, Int. Conference on Computer Vision Theory and Applications*, Barcelona, Spain, March 8-11, 2007.